

## **Stručni studij**

# **Hotel**

**Projektni zadatak iz kolegija  
Baze podataka**

## Sadržaj

1. Opis projektnog zadatka .....	1
2. ER dijagram.....	2
3. Relacijski model .....	3
4. Opis odabranog rješenja .....	4
5. SQL naredbe za kreiranje baze.....	5
6. Punjenje baze podacima .....	8
7. Upiti koji se najčešće koriste .....	9
8. Zaključak .....	12
REFERENCE .....	13
DODATAK .....	13

Marenić

# 1. Opis projektnog zadatka

## “Hotel”

### Opis:

Potrebno je osmisliti i realizirati bazu podataka koja može učinkovito poslužiti za potrebe recepcije hotela. U fazi analize, utvrđeno je da korisnik želi imati ažurnu evidenciju soba (broj sobe, broj ležajeva, tip kupaone, TV, bar, ...), rezervacija, gostiju i njihovih računa (hrana, piće, hotelske usluge,...) za period boravka u hotelu. Osnovni zahtjevi korisnika su slijedeći:

- omogućiti uvid u trenutno slobodne i/ili zauzete sobe
- prikaz obavljenih telefonskih razgovora iz sobe
- prikaz troškova po gostu
- prikaz stanja rezervacija i slobodnih soba

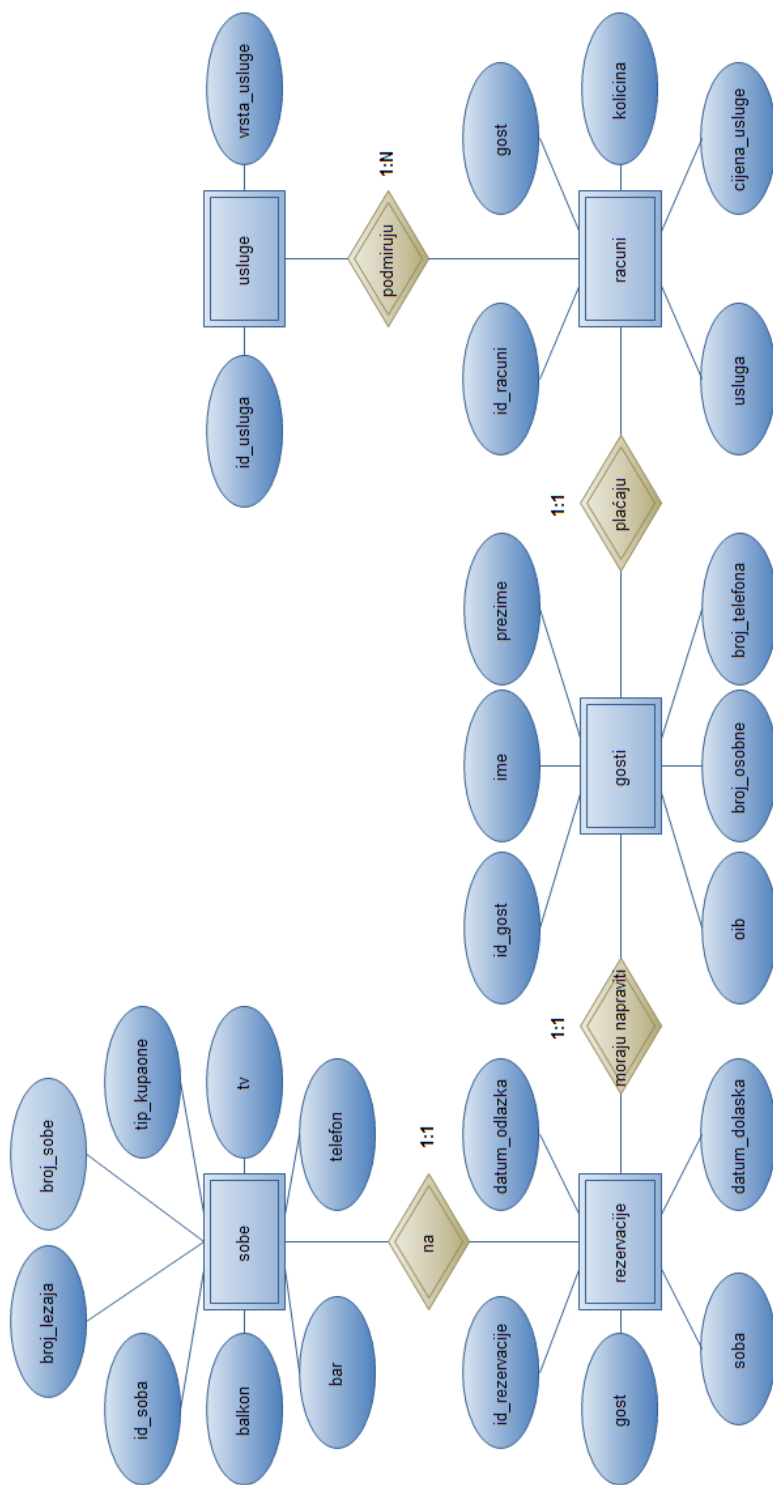
Baza podataka treba biti u što većoj mjeri normalizirana.

### Zadaci:

1. Napraviti ER dijagram (entity-relationship diagram) dijagram sa svim bitnim elementima.
2. Napraviti relacijski model
3. Dati kraći tekstualni opis odabranog rješenja uz osvrt na eventualne specifičnosti, pretpostavke ili ograničenja u modelu.
4. Napraviti SQL naredbe za kreiranje baze podataka koja odgovara relacijskom modelu.
5. Napraviti SQL naredbe kojima se baza puni podacima za potrebe testiranja.
6. Napraviti primjer SQL upita za koje se očekuje da će biti najčešće upotrebljavani od strane korisnika baze podataka (prema zahtjevima u opisu zadatka), opisati ih riječima (koja je svrha upita?) i dati konkretni primjer rezultata kakav se dobije takvim upitom



## 2. ER dijagram



Slika 1. ER dijagram baze Hotel

### 3. Relacijski model

#### gosti

Polje	Vrsta	Null	Zadano	Poveznica
id_gost	int(11)	No		
ime	varchar(30)	No		
prezime	varchar(50)	No		
oib	int(11)	No		
broj_osobne	int(9)	No		
broj_telefona	int(15)	No		

#### racuni

Polje	Vrsta	Null	Zadano	Poveznica
id_racuni	int(11)	No		
gost	int(11)	No		gosti -> id_gost
usluga	int(11)	No		usluge -> id_usluga
kolicina	int(11)	No		
cijena_usluge	float(10,2)	No		

#### rezervacije

Polje	Vrsta	Null	Zadano	Poveznica
id_rezervacije	int(11)	No		
gost	int(11)	No		gosti -> id_gost
soba	int(11)	No		sobe -> id_soba
datum_dolaska	date	No		
datum_odlaska	date	No		

#### sobe

Polje	Vrsta	Null	Zadano	Poveznica
id_soba	int(11)	No		
broj_sobe	char(4)	No		
broj_lezaja	int(2)	No		
tip_kupaone	varchar(30)	No		
balkon	char(2)	No	DA	
tv	char(2)	No	DA	
bar	char(2)	No	DA	
telefon	char(2)	No	DA	

#### usluge

Polje	Vrsta	Null	Zadano	Poveznica
id_usluga	int(11)	No		
vrsta_usluge	varchar(50)	No		

## 4. Opis odabranog rješenja

Baza je napravljena u MySQL-u (XAMPP). Udovoljava svim postavljenim uvjetima zadanim projektnim zadatkom. Sastoji se od 5 tablica.

### ✓ **tablica gosti**

Tablica unos gostiju. Uz unos imena i prezimena, treba unijeti osobne podatke kao što je OIB, broj osobne iskaznice i broj telefona (fiksni ili mobilni). Kao primarni ključ postavljen je id\_gost. To je ujedno i strani ključ na tablice racuni i rezervacije.

### ✓ **tablica racuni**

Ova tablica sadrži podatke o izdanim računima za pojedinog gosta i uz vezanu uslugu. Za primarni ključ postavljen je id\_racuni. Tablica je ograničena sa dva strana ključa: id\_gost i id\_usluga.

### ✓ **tablica rezervacije**

Tablica za unos rezervacije sa poljima: gost, soba, datum\_dolaska, datum\_odlaska. Ona je ograničena sa dva strana ključa: id\_gost i id\_soba.

### ✓ **tablica sobe**

Tablica za unos sobe prema broju i pripadajućom opremljenošću sobe. Tako se može izdvojiti soba sa više ležaja, prema tipu kupaonice (tuš, kada...), balkon ili da ima pogodnosti kao što je televizija, bar sa pićem koji se dodatno naplaćuje i telefon. Taj dodatni sadržaj je postavljen na zadano DA, tako da ako soba ne sadrži nešto od toga potrebno je upisati NE. Primarni ključ je id\_soba, a to je ujedno i strani ključ na tablicu rezervacije.

### ✓ **tablica usluge**

Mala tablica za unos svih usluga koje nudi hotel. Sastoji se od id\_usluga i stupca vrsta\_usluge. Primarni ključ ujedno je i strani ključ na tablicu racuni.

## 5. SQL naredbe za kreiranje baze

### Kreiranje baze podataka:

```
CREATE DATABASE `hotel` DEFAULT CHARACTER SET latin1 COLLATE  
latin1_swedish_ci;  
USE `hotel`;
```

### Kreiranje tablice >>gosti<<

```
CREATE TABLE IF NOT EXISTS `gosti` (  
  `id_gost` int(11) NOT NULL AUTO_INCREMENT,  
  `ime` varchar(30) COLLATE utf8_bin NOT NULL,  
  `prezime` varchar(50) COLLATE utf8_bin NOT NULL,  
  `oib` char(11) COLLATE utf8_bin NOT NULL,  
  `broj_osobne` int(9) NOT NULL,  
  `broj_telefona` int(15) NOT NULL,  
  PRIMARY KEY (`id_gost`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=4 ;
```

### Kreiranje tablice >>racuni<<

```
CREATE TABLE IF NOT EXISTS `racuni` (  
  `id_racuni` int(11) NOT NULL AUTO_INCREMENT,  
  `gost` int(11) NOT NULL,  
  `usluga` int(11) NOT NULL,  
  `kolicina` int(11) NOT NULL,  
  `cijena_usluge` float(10,2) NOT NULL,  
  PRIMARY KEY (`id_racuni`),  
  KEY `gost` (`gost`),  
  KEY `usluga` (`usluga`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=4 ;
```

## **Kreiranje tablice >>rezervacije<<**

```
CREATE TABLE IF NOT EXISTS `rezervacije` (  
  `id_rezervacije` int(11) NOT NULL AUTO_INCREMENT,  
  `gost` int(11) NOT NULL,  
  `soba` int(11) NOT NULL,  
  `datum_dolaska` date NOT NULL,  
  `datum_odlaska` date NOT NULL,  
  PRIMARY KEY (`id_rezervacije`),  
  KEY `gost` (`gost`),  
  KEY `soba` (`soba`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=2 ;
```

## **Kreiranje tablice >>sobe<<**

```
CREATE TABLE IF NOT EXISTS `sobe` (  
  `id_soba` int(11) NOT NULL AUTO_INCREMENT,  
  `broj_sobe` int(11) NOT NULL,  
  `broj_lezaja` int(2) NOT NULL,  
  `tip_kupaone` varchar(30) COLLATE utf8_bin NOT NULL,  
  `balkon` char(2) COLLATE utf8_bin NOT NULL DEFAULT 'DA',  
  `tv` char(2) COLLATE utf8_bin NOT NULL DEFAULT 'DA',  
  `bar` char(2) COLLATE utf8_bin NOT NULL DEFAULT 'DA',  
  `telefon` char(2) COLLATE utf8_bin NOT NULL DEFAULT 'DA',  
  PRIMARY KEY (`id_soba`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=5 ;
```

## **Kreiranje tablice >>usluge<<**

```
CREATE TABLE IF NOT EXISTS `usluge` (  
  `id_usluga` int(11) NOT NULL AUTO_INCREMENT,  
  `vrsta_usluge` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_usluga`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```



## **Postavljanje ograničenja**

```
ALTER TABLE `racuni`
```

```
ADD CONSTRAINT `racuni_ibfk_3` FOREIGN KEY (`usluga`) REFERENCES `usluge`  
(`id_usluga`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `racuni_ibfk_4` FOREIGN KEY (`gost`) REFERENCES `gosti`  
(`id_gost`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Ograničenje na tablici „racuni“. Vidi se da postoje dva strana ključa: id\_usluga i id\_gost.

```
ALTER TABLE `rezervacije`
```

```
ADD CONSTRAINT `rezervacije_ibfk_1` FOREIGN KEY (`soba`) REFERENCES `sobe`  
(`id_soba`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `rezervacije_ibfk_2` FOREIGN KEY (`gost`) REFERENCES `gosti`  
(`id_gost`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Ograničenje na tablici „rezervacije“. Postoje dva strana ključa: id\_soba i id\_gost.

## 6. Punjenje baze podacima

### Unos podataka u tablicu >>gosti<<

```
INSERT INTO `GOSTI` (`ID_GOST`, `IME`, `PREZIME`, `OIB`, `BROJ_OSOBNE`,  
`BROJ_TELEFONA`) VALUES  
(1, 'JOSIP', 'MARIC', '21474836474', 589315487, 918569366),  
(2, 'IVAN', 'IVIC', '78256359845', 471259585, 997896325),  
(3, 'MARIJA', 'JURIC', '56528654475', 156638956, 31775921);
```

### Unos podataka u tablicu >>racuni<<

```
INSERT INTO `RACUNI` (`ID_RACUNI`, `GOST`, `USLUGA`, `KOLICINA`,  
`CIJENA_USLUGE`) VALUES  
(1, 1, 5, 16, 1960.00),  
(2, 2, 1, 1, 135.00),  
(3, 3, 3, 2, 339.00);
```

### Unos podataka u tablicu >>rezervacije<<

```
INSERT INTO `REZERVACIJE` (`ID_REZERVACIJE`, `GOST`, `SOBA`, `DATUM_DOLASKA`,  
`DATUM_ODLASKA`) VALUES  
(1, 1, 2, '2011-02-06', '2011-02-22');
```

### Unos podataka u tablicu >>sobe<<

```
INSERT INTO `SOBE` (`ID_SOBA`, `BROJ_SOBE`, `BROJ_LEZAJA`, `TIP_KUPAONE`,  
`BALKON`, `TV`, `BAR`, `TELEFON`) VALUES  
(1, 1A, 2, 'KADA', 'DA', 'DA', 'DA', 'DA'),  
(2, 2B, 1, 'JACUZZI', 'NE', 'DA', 'DA', 'DA'),  
(3, 15A, 4, 'TUS', 'DA', 'NE', 'NE', 'NE'),  
(4, 10B, 2, 'TUS', 'DA', 'DA', 'NE', 'DA');
```

### Unos podataka u tablicu >>usluge<<

```
INSERT INTO `usluge` (`id_usluga`, `vrsta_usluge`) VALUES  
(1, 'masaza'),  
(2, 'nocenje'),  
(3, 'nocenje + vecera'),  
(4, 'pice'),  
(5, 'rezervacija');
```

## 7. Upiti koji se najčešće koriste

### Ispis podataka iz tablice >>racuni<<

```
SELECT * FROM racuni;
```

Naredbom SELECT \* (ALL) prikazuju se svi podaci u tablici racuni. Ovo nije najbolji prikaz pa ćemo koristiti slijedeći izraz:

```
SELECT `gosti`.`ime`, `gosti`.`prezime`, `usluge`.`vrsta_usluge`, `racuni`.`kolicina`,  
`racuni`.`cijena_usluge`  
FROM `racuni`, `gosti`, `usluge`  
WHERE racuni.gost = gosti.id_gost AND racuni.usluga = usluge.id_usluga
```

Ovako smo dobili lijepi prikaz. Stupcima „gost“ i „usluga“ dodijelili smo prave vrijednosti koje se pomoću ograničenja dodijeljene prema određenom ID-u.

Pošto se dohvaća više tablica, a u slučaju da u njima postoji isti naziv stupaca, koristi se slijedeći izraz da bi se ispravno pristupilo podacima: `tablica`.`ime\_stupca`.

### Prihod hotela

```
SELECT SUM(kolicina * cijena_usluge) AS 'Ukupni prihod' FROM racuni;
```

Da bi smo provjerili koliki je ukupni prihod koristimo naredbu SUM. Pomoću nje zbrojili smo sve vrijednosti koje se nalaze u stupcima „kolicina“ i „cijena\_usluge“.

### Potrošeno na razgovore

```
SELECT SUM(kolicina*cijena_usluge) AS 'Ukupno telefona' FROM racuni  
WHERE racuni.usluga = 6
```

Slično kao i izraz gore ali u ovom slučaju želimo izračunati koliki je napraviti račun od samog telefoniranja. To smo postigli tako da smo postavili uvjet WHERE koji je pod brojem 6, odnosno iza toga broja pomoću stranog ključa nalazi se oznaka telefon. Na taj način se zbrajaju samo oni redovi čija je oznaka usluge pod brojem 6.

### Prikaz troška za jednog gosta

```
SELECT SUM(kolicina*cijena_usluge) AS 'Ukupno za platiti' FROM racuni  
WHERE gost = 1;
```

Na ovaj način dobili smo prikaz troškova koji se odnose na jednog gosta. U ovom slučaju to je gost pod id brojem 1. Sa AS stavili smo da ljepše prikaže ime stupca.

## **Ispis podataka iz tablice >>rezervacije<<**

```
SELECT `gosti`.`ime`, `gosti`.`prezime`, `sobe`.`broj_sobe`,  
`rezervacije`.`datum_dolaska`, `rezervacije`.`datum_odlaska`  
FROM `rezervacije`, `gosti`, `sobe`  
WHERE rezervacije.gost = gosti.id_gost AND rezervacije.soba=sobe.id_soba
```

Ovaj izraz daje lijepi prikaz. Pomoću naredbe WHERE zamijenili smo vrijednosti koji su kriju iza gosti.ime, gosti.prezime i sobe.broj\_sobe.

## **Ispis podatka iz tablice >>gosti<<**

```
SELECT `gosti`.`ime`, `gosti`.`prezime`, `gosti`.`oib`, `gosti`.`broj_osobne`,  
`gosti`.`broj_telefona`  
FROM `gosti`
```

Ovim upitom dohvaćaju se svi podaci u tablici „gosti“.

## **Ispis podataka iz tablice >>sobe<<**

### **Popis svih soba**

```
SELECT `sobe`.`broj_sobe`, `sobe`.`broj_lezaja`, `sobe`.`tip_kupaone`, `sobe`.`balkon`,  
`sobe`.`tv`, `sobe`.`bar`, `sobe`.`telefon`  
FROM `sobe`
```

Ovim upitom dohvaćaju se svi stupci iz tablice „sobe“.

### **Zauzete sobe**

```
SELECT DISTINCT broj_sobe, broj_lezaja, tip_kupaone, balkon, tv, bar, telefon FROM  
sobe JOIN rezervacije WHERE rezervacije.soba = sobe.id_soba
```

Upit nam daje podatke o zauzetim sobama.

### **Slobodne sobe**

```
SELECT DISTINCT broj_sobe, broj_lezaja, tip_kupaone, balkon, tv, bar, telefon FROM  
sobe LEFT JOIN rezervacije ON (sobe.id_soba=rezervacije.soba) WHERE  
rezervacije.id_rezervacije IS NULL
```

Upit za ispis svih soba koje nisu zauzete.

## **Ispis podataka iz tablice >>usluge<<**

```
SELECT * FROM usluge;
```

Ovo je najbolji i najjednostavniji način za ispis svih podataka iz tablice „usluge“. Tablica je mala pa nije potrebno postavljati dodatne upite.

## **Izmjena podatka u tablici >>rezervacije<<** **- izmjena datuma odlaska**

```
UPDATE `hotel`.`rezervacije` SET `soba` = '4',  
`datum_odlaska` = '2011-02-26' WHERE `rezervacije`.`id_rezervacije` =3 LIMIT 1 ;
```

Naredbom update mijenjamo podatke u tablici letovi prema id-u 3. LIMIT znači da je ograničeno brisanje samo na jedan red u tablici.

## **Brisanje podatka u tablici >>gost<<** **- gost više ne koristi usluge**

```
DELETE FROM `hotel`.`gosti` WHERE `gosti`.`id_gost` = 4 LIMIT 1;
```

Pomoću naredbe DELETE brišemo iz tablice red s pripadajućim primarnim ključem 4.

## 8. Zaključak

Ovaj projekt smo izradili pomoću MySQL baze, koja se nalazi u paketu XAMPP. Naučili smo kako baza radi i neke osnovne naredbe kao što je CREATE, SELECT, UPDATE, SUM... Za tip baze odabrali smo InnoDB. On nam omogućava stvaranje ograničenja na tablicama.

Baza udovoljava većini uvjeta koji su postavljeni projektnim zadatkom. Također je dosta optimizirana što je čini vrlo malom i jednostavnom za korištenje.

ER dijagram napravili smo pomoću programa Edraw UML Diagram 5.1, a provjeru uvjeta smo odradili u phpmyadminu koji se nalazi unutar XAMPP paketa.

Marenić

## REFERENCE

- Google, web tražilica
- Laboratorijske vježbe, kolegij Baze podataka
- PHP i MySQL, W. Jason Gilmore, Apress, knjiga
- W3schools, <http://www.w3schools.com/sql/default.asp>, tutoriali za SQL
- MySQL, <http://dev.mysql.com/doc/refman/5.0/en/tutorial.html>, službeni tutorial za zadnju verziju mysqla

## DODATAK

- Baza [..\mysql\hotel.sql](#) koja se nalazi u mapi mysql

Marenić